INCREASING TEAM EFFICIENCY WITH STORYBOOKJS

BY ALEX WILSON

WHAT IS STORYBOOK?

INTERACTIVE DEMO ENVIRONMENT OF ALL UI COMPONENTS

ALLOWS A USER TO RUN AND VIEW A PIECE OF FUNCTIONALITY IN AN **ISOLATED ENVIRONMENT**

REQUIRES **MINIMAL CONFIGURATION** FOR EACH COMPONENT BUILT

CUSTOMIZABLE TOOLING TO ALLOW FOR NEW STORYBOOK FEATURES

DEPLOYABLE APPLICATION

OPEN SOURCE FREE SOFTWARE

WHY SHOULD YOU USE STORYBOOK

To increase the feedback loop

To provide a platform to display more documentation about how to use your components

To develop components faster with hot reloading

To Increase the speed of team member onboarding

To get more visibility into the “what if” scenarios of a component view

WHAT FRAMEWORKS DOES STORYBOOK SUPPORT?

Angular

Ember

Riot

React (and native)

HTML5

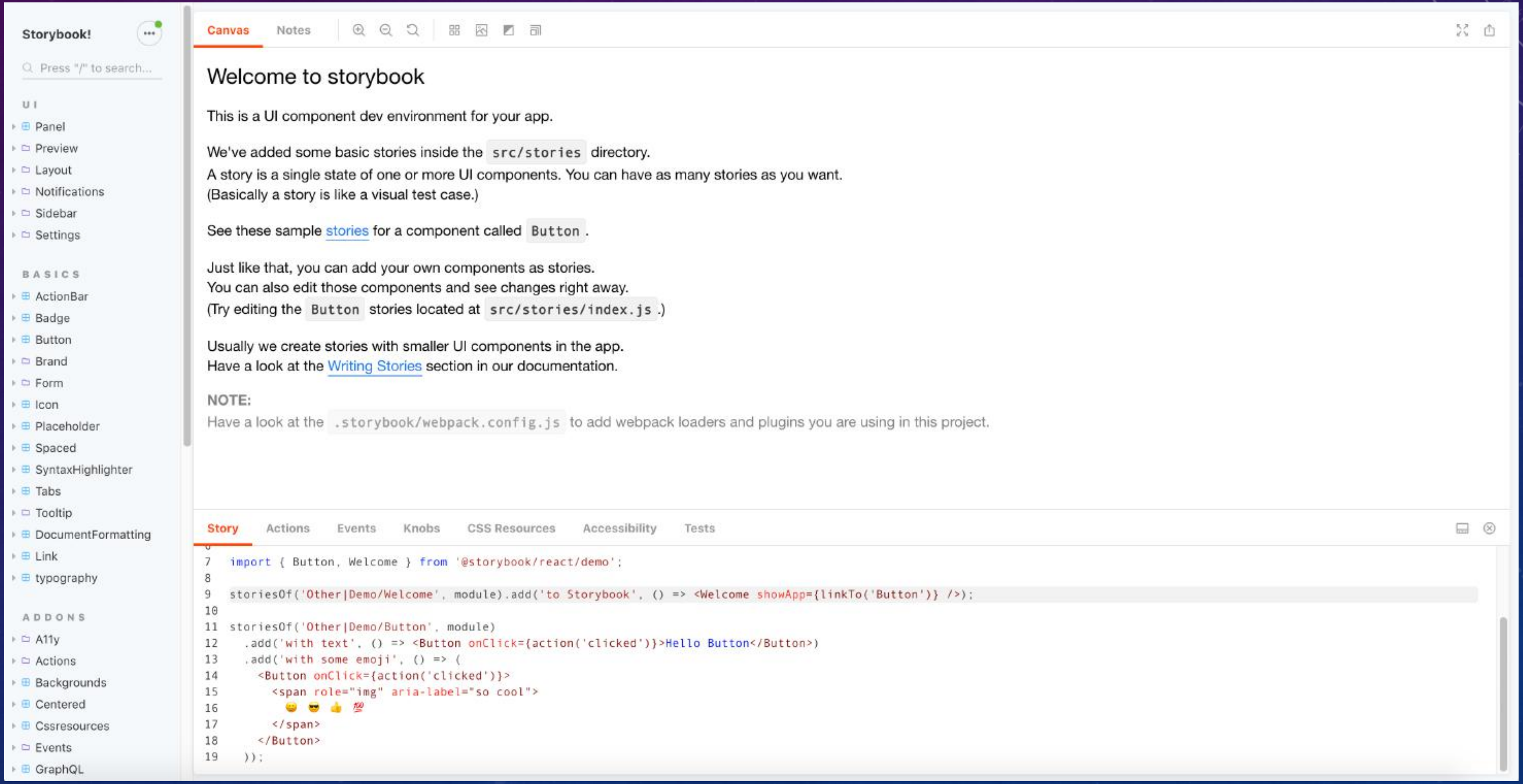
Preact

Vue

Svelte

Mithril

STORYBOOK LAYOUT



Welcome to storybook

This is a UI component dev environment for your app.

We've added some basic stories inside the `src/stories` directory.
A story is a single state of one or more UI components. You can have as many stories as you want.
(Basically a story is like a visual test case.)

See these sample stories for a component called `Button`.

Just like that, you can add your own components as stories.
You can also edit those components and see changes right away.
(Try editing the `Button` stories located at `src/stories/index.js`.)

Usually we create stories with smaller UI components in the app.
Have a look at the Writing Stories section in our documentation.

NOTE:
Have a look at the `.storybook/webpack.config.js` to add webpack loaders and plugins you are using in this project.

```
import { Button, Welcome } from '@storybook/react/demo';

storiesOf('Other|Demo/Welcome', module).add('to Storybook', () => <Welcome showApp={linkTo('Button')} />);

storiesOf('Other|Demo/Button', module)
  .add('with text', () => <Button onClick={action('clicked')}>Hello Button</Button>)
  .add('with some emoji', () => (
    <Button onClick={action('clicked')}>
      <span role="img" aria-label="so cool">
        😀 😎 👍 💯
      </span>
    </Button>
  ));
```

FEATURES

Interactive knobs

Viewport manipulation

Accessibility testing

Colorblind visibility testing

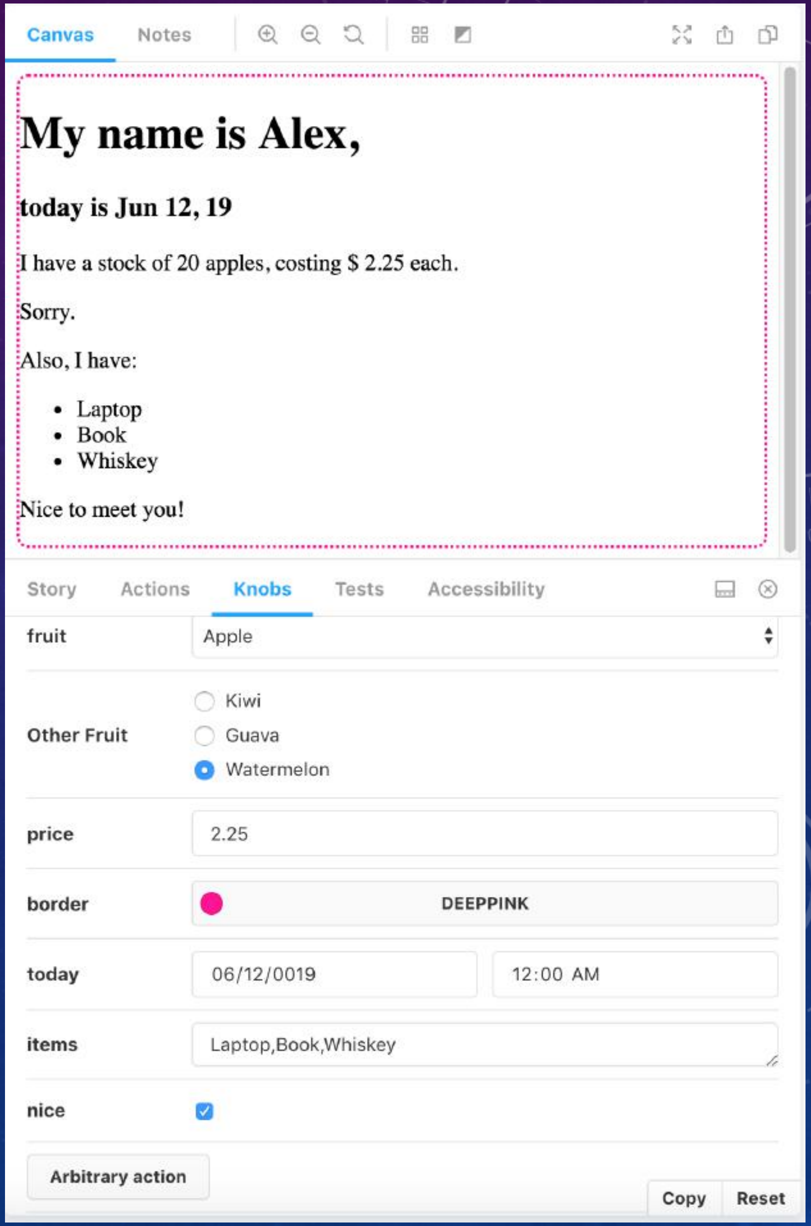
Background color switching

Jest unit test view

Deployable static demo application

FEATURES: INTERACTIVE KNOBS

- Allows the user to edit a component's property data using the Storybook UI
- Changes to the data cause a native re-rendering of the component immediately without refresh
- The data configuration can be copied and sent out as a URL to other members of your team for easy replication of scenarios
- Easy to find bugs by entering in data that would not be commonly seen from your website



My name is Alex,

today is Jun 12, 19

I have a stock of 20 apples, costing $ 2.25 each.

Sorry.

Also, I have:

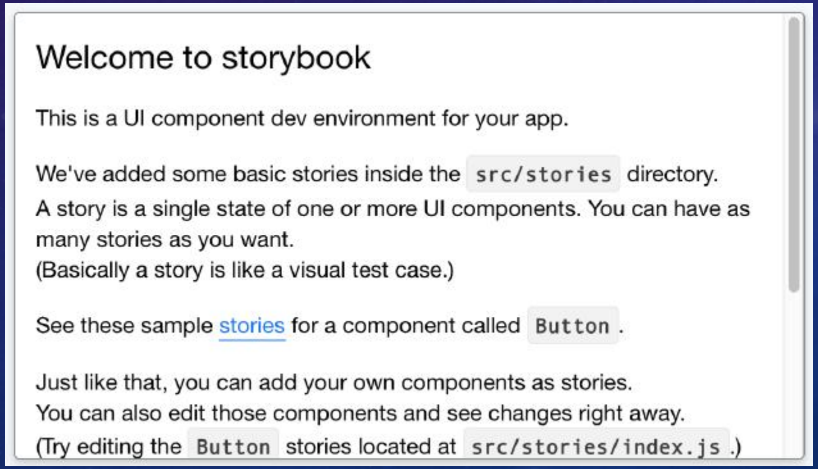
- Laptop
- Book
- Whiskey

Nice to meet you!

fruit	Apple
Other Fruit	○ Kiwi ○ Guava [✓ Watermelon]
price	2.25
border	● DEEPPINK
today	06/12/0019 12:00 AM
items	Laptop,Book,Whiskey
nice	✓

FEATURES: VIEWPORT MANIPULATION

- Gives the user the ability to view the given component at the device level
- Default or customized sizes can be used based on a configuration
- Options to rotate the view to either portrait or landscape
- Helpful for mobile responsiveness validation



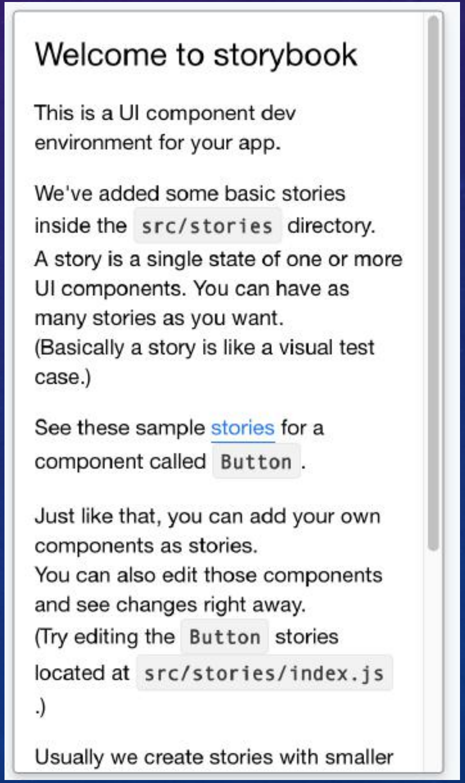
Welcome to storybook

This is a UI component dev environment for your app.

We've added some basic stories inside the `src/stories` directory.
A story is a single state of one or more UI components. You can have as many stories as you want.
(Basically a story is like a visual test case.)

See these sample stories for a component called `Button`.

Just like that, you can add your own components as stories.
You can also edit those components and see changes right away.
(Try editing the `Button` stories located at `src/stories/index.js`.)



Welcome to storybook

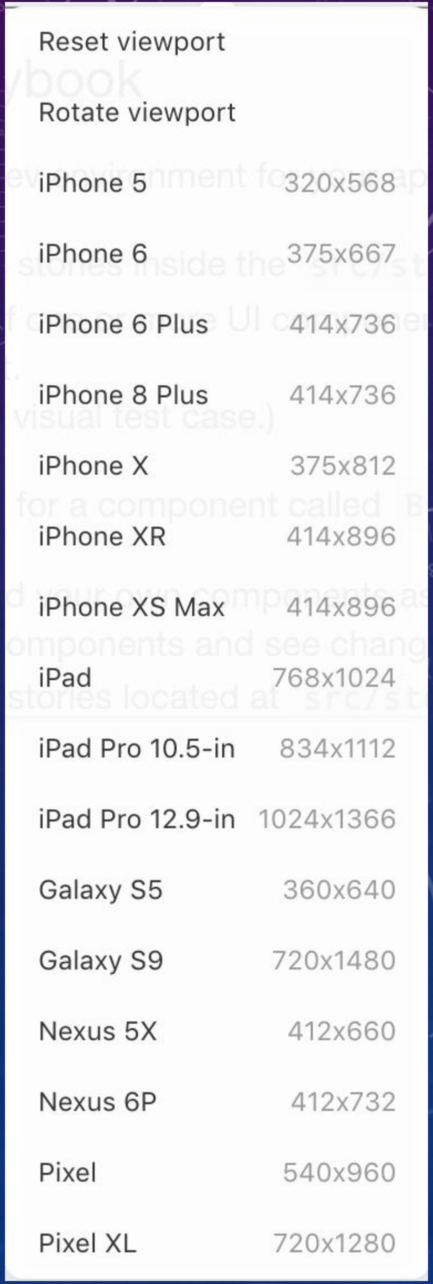
This is a UI component dev environment for your app.

We've added some basic stories inside the `src/stories` directory.
A story is a single state of one or more UI components. You can have as many stories as you want.
(Basically a story is like a visual test case.)

See these sample stories for a component called `Button`.

Just like that, you can add your own components as stories.
You can also edit those components and see changes right away.
(Try editing the `Button` stories located at `src/stories/index.js`.)

Usually we create stories with smaller



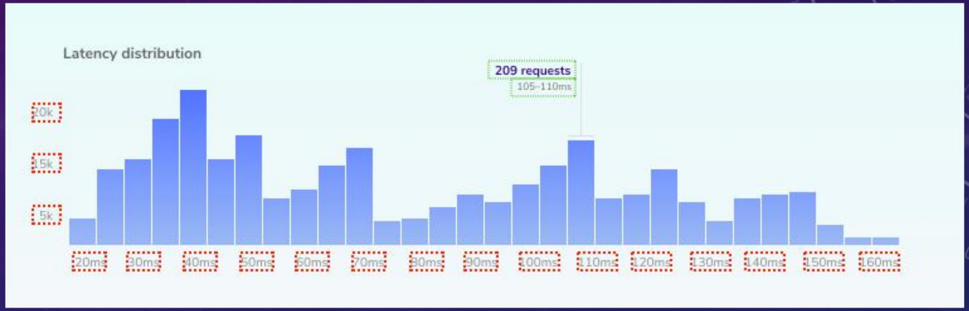
Reset viewport

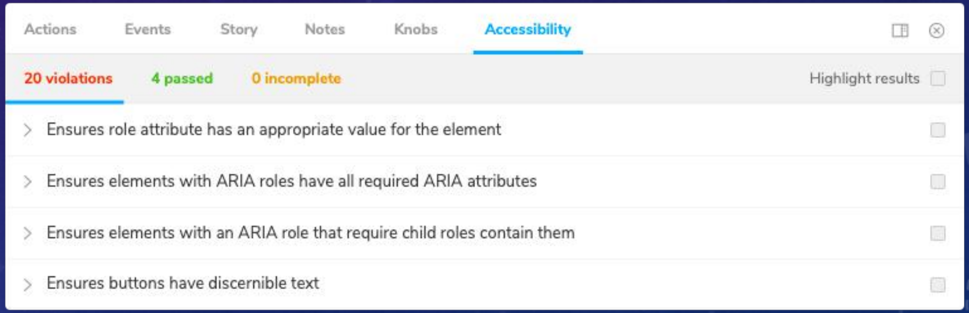
Rotate viewport

iPhone 5	320x568
iPhone 6	375x667
iPhone 6 Plus	414x736
iPhone 8 Plus	414x736
iPhone X	375x812
iPhone XR	414x896
iPhone XS Max	414x896
iPad	768x1024
iPad Pro 10.5-in	834x1112
iPad Pro 12.9-in	1024x1366
Galaxy S5	360x640
Galaxy S9	720x1480
Nexus 5X	412x660
Nexus 6P	412x732
Pixel	540x960
Pixel XL	720x1280

FEATURES: ACCESSIBILITY TESTING

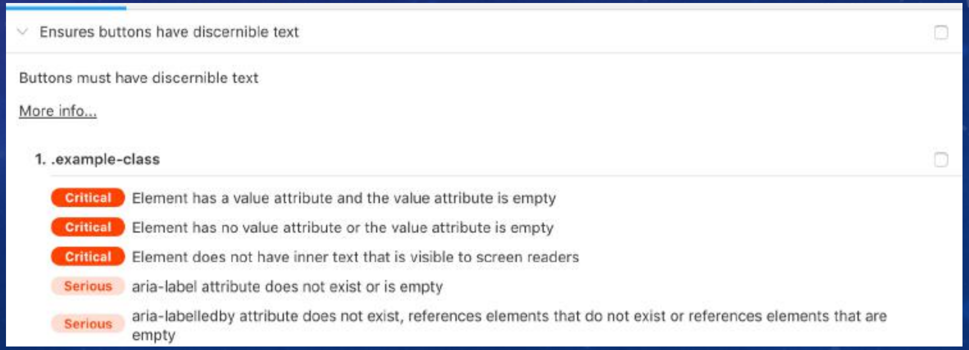
- Axe accessibility driven accessibility testing
- Provides a way to highlight issues
- Displays number of violations and passes
- Describes what passed and what failed as well as provides the element trace that is causing the failure
- A “more info” link is displayed under the rule which will send the user to a more in-depth explanation





20 violations 4 passed 0 incomplete Highlight results

- Ensures role attribute has an appropriate value for the element
- Ensures elements with ARIA roles have all required ARIA attributes
- Ensures elements with an ARIA role that require child roles contain them
- Ensures buttons have discernible text



Ensures buttons have discernible text

Buttons must have discernible text

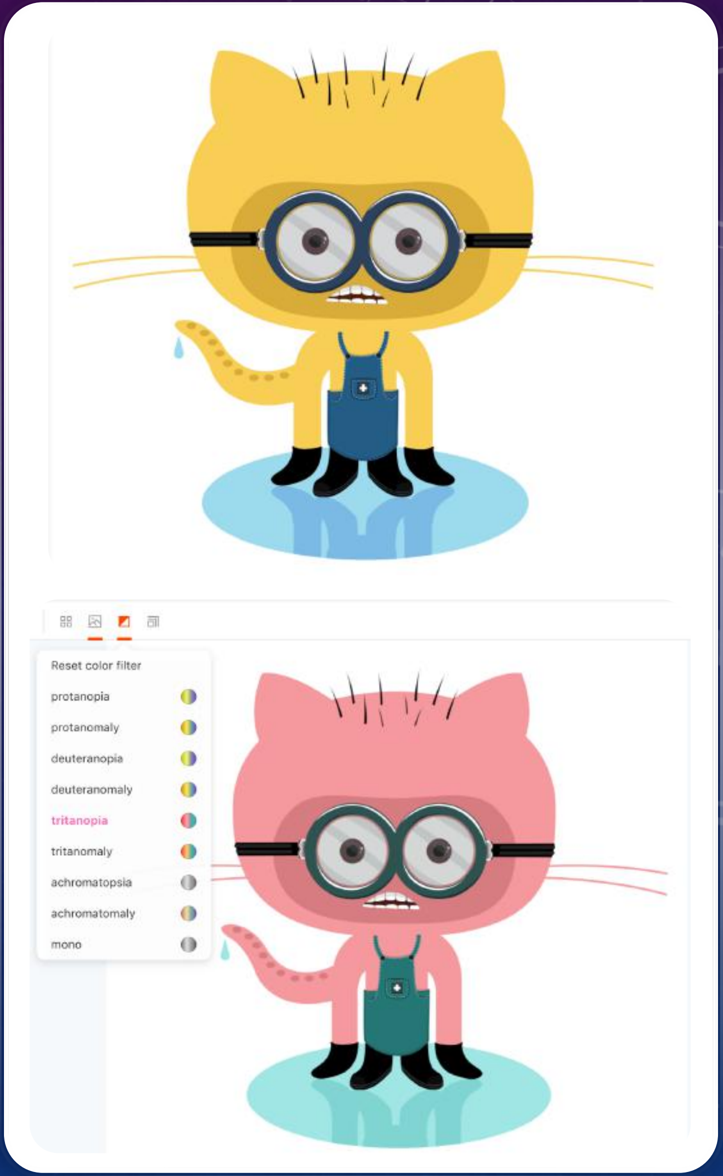
More info...

1. .example-class

Critical	Element has a value attribute and the value attribute is empty
Critical	Element has no value attribute or the value attribute is empty
Critical	Element does not have inner text that is visible to screen readers
Serious	aria-label attribute does not exist or is empty
Serious	aria-labelledby attribute does not exist, references elements that do not exist or references elements that are empty

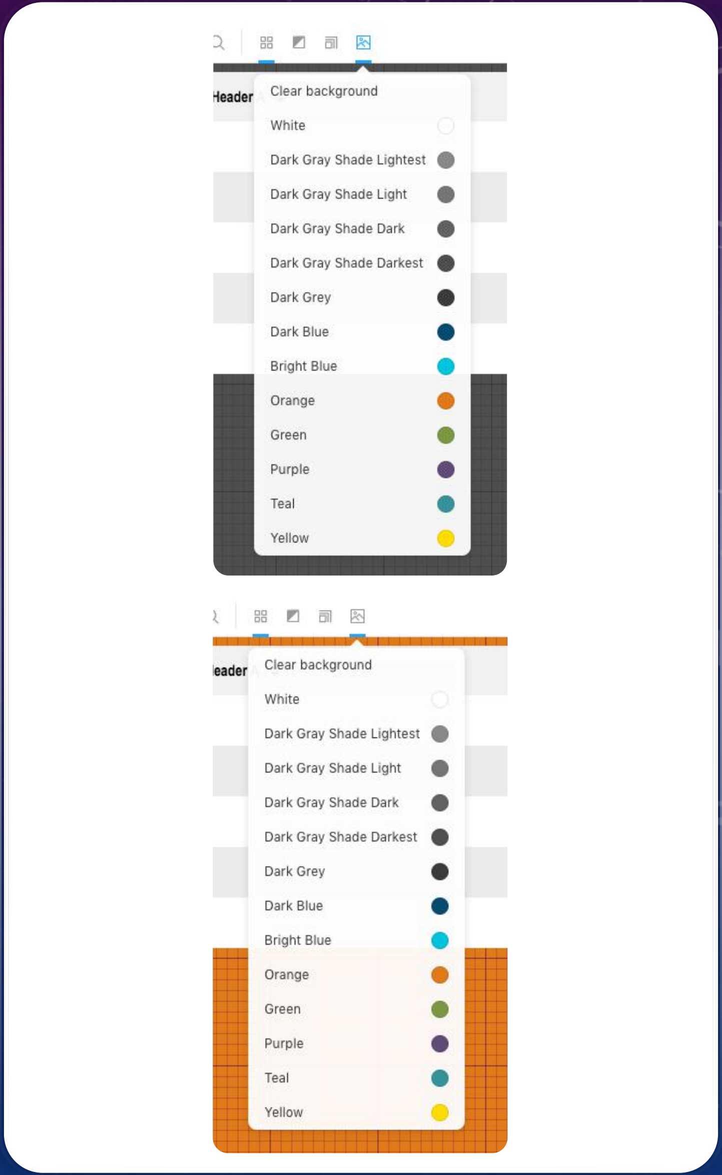
FEATURES: COLORBLIND VISIBILITY TESTING

- Emulates what someone with color blindness would see when looking at a component
- Visual deficiency options
  - Protanopia (red-green color blindness)
  - Protanomaly (reduced sensitivity to red light)
  - Deuteranopia (green color blindness)
  - Deuteranomaly (reduced sensitivity to green light)
  - Tritanopia (blue yellow color blindness)
  - Tritanomaly (reduced sensitivity to blue light)
  - Achromatomaly and Achromatopsia (reduced or partial color blindness)
  - Mono (no color)



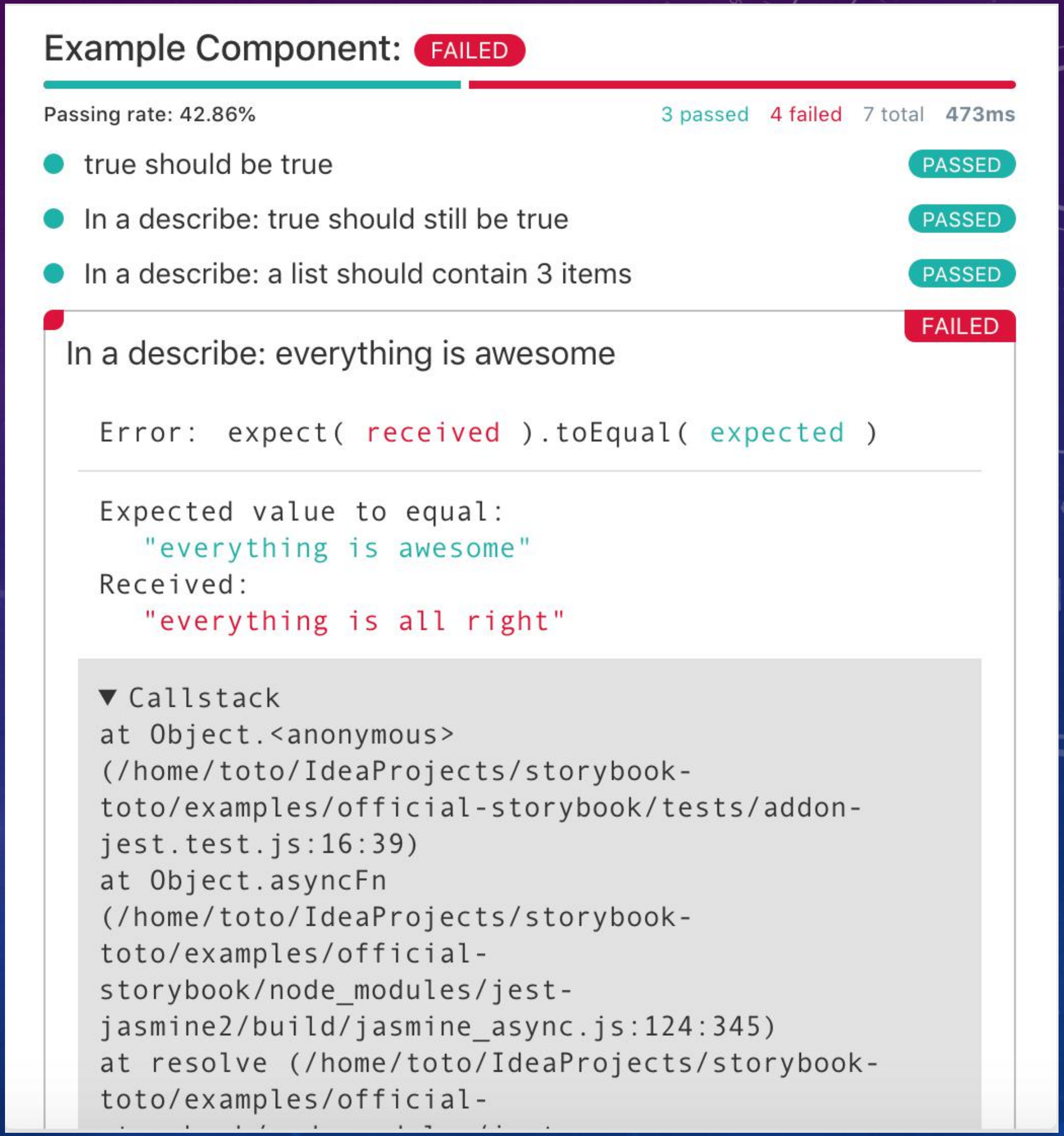
FEATURES: BACKGROUND COLOR SWITCHING

- Allows the user to switch the background behind the component
- Creates a visual representation of what a component will look like in different environments



FEATURE: JEST TEST VIEW

- Displays all unit tests for a single component
- Provides a stack trace for the failing tests
- Hot reloading of the Jest view during development which makes it easy to determine breaking changes
- Calculates a percentage passing rate and color



Example Component: FAILED

Passing rate: 42.86% 3 passed 4 failed 7 total 473ms

- true should be true PASSED
- In a describe: true should still be true PASSED
- In a describe: a list should contain 3 items PASSED

FAILED

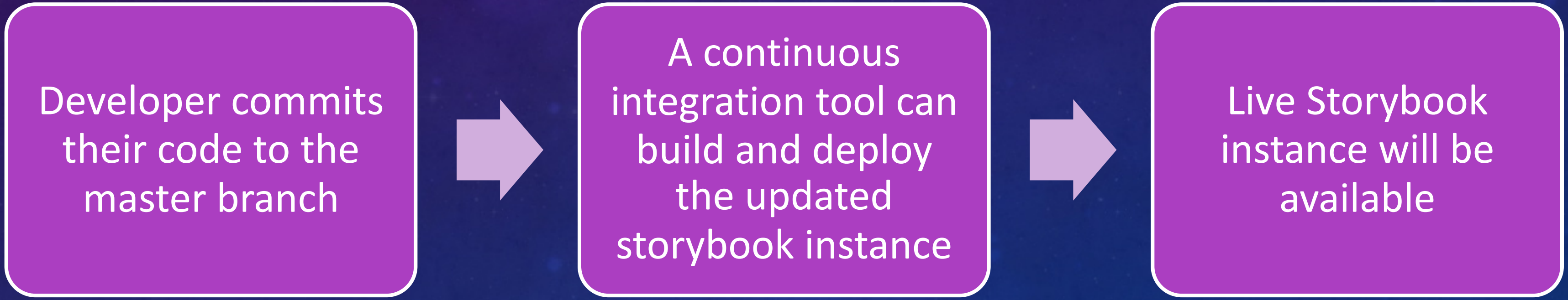
In a describe: everything is awesome

```
Error:  expect( received ).toEqual( expected )

Expected value to equal:
  "everything is awesome"
Received:
  "everything is all right"

▼ Callstack
at Object.<anonymous>
(/home/toto/IdeaProjects/storybook-
toto/examples/official-storybook/tests/addon-
jest.test.js:16:39)
at Object.asyncFn
(/home/toto/IdeaProjects/storybook-
toto/examples/official-
storybook/node_modules/jest-
jasmine2/build/jasmine_async.js:124:345)
at resolve (/home/toto/IdeaProjects/storybook-
toto/examples/official-
```

FEATURES: DEPLOYABLE STATIC DEMO APPLICATION

Storybook demoes can be built as a stand alone application with a URL that can be used to show any member of your team what the most recently completed components look like

*This is great for teams that build out libraries that contain components that may not show up on the website for a stretch of time


Developer commits their code to the master branch

A continuous integration tool can build and deploy the updated storybook instance

Live Storybook instance will be available

FOLLOW ME ON

Twitter: @CodeByAlex

GitHub: @CodeByAlex

My website: CodeByAlex.com

DEMO